

# BellaBeat\_\_R-Markdown\_v2

Lauren Tran

2022-11-30

## Business Tasks

How customers use smart devices through smart device usage data, so Bellabeat can introduce new marketing strategy.

1. What are some trends in smart device usage?
2. How could these trends apply to Bellabeat customers?
3. How could these trends help influence Bellabeat marketing strategy?

## Stakeholders

- Urška Sršen: Bellabeat's cofounder
- Sando Mur: Mathematician and Bellabeat's cofounder
- Other people in Bellabeat's analytics team

## Process & Analyze the data

1. Setting up the environment and loading the datasets.

```
library(tidyverse)
```

```
## -- Attaching packages ----- tidyverse 1.3.2 --
## v ggplot2 3.4.0      v purrr   0.3.5
## v tibble  3.1.8      v dplyr  1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.3      v forcats 0.5.2
## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(lubridate)
```

```
## Loading required package: timechange
##
## Attaching package: 'lubridate'
##
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(dplyr)
library(ggplot2)
library(tidyr)
library(tibble)
```

```
activity <- read_csv("C:\\Users\\laure\\OneDrive\\Documents\\Data Study\\Case Study\\Fitabase Data 4.12.16\\activity.csv")
```

```
## Rows: 934 Columns: 16
## -- Column specification -----
## Delimiter: ","
## dbl (15): Id, Total_intensity, TotalSteps, TotalDistance, TrackerDistance, ...
## date (1): ActivityDate
##
## i Use 'spec()' to retrieve the full column specification for this data.
## i Specify the column types or set 'show_col_types = FALSE' to quiet this message.
```

```
hourly_intensity <- read_csv("C:\\Users\\laure\\OneDrive\\Documents\\Data Study\\Case Study\\Fitabase Data 4.12.16\\hourly_intensity.csv")
```

```
## Rows: 22099 Columns: 5
## -- Column specification -----
## Delimiter: ","
## chr (1): ActivityDate
## dbl (3): Id, TotalIntensity, AverageIntensity
## time (1): ActivityHour
##
## i Use 'spec()' to retrieve the full column specification for this data.
## i Specify the column types or set 'show_col_types = FALSE' to quiet this message.
```

```
sleep <- read_csv("C:\\Users\\laure\\OneDrive\\Documents\\Data Study\\Case Study\\Fitabase Data 4.12.16\\sleep.csv")
```

```
## Rows: 413 Columns: 5
## -- Column specification -----
## Delimiter: ","
## chr (1): SleepDay
## dbl (4): Id, TotalSleepRecords, TotalMinutesAsleep, TotalTimeInBed
##
## i Use 'spec()' to retrieve the full column specification for this data.
## i Specify the column types or set 'show_col_types = FALSE' to quiet this message.
```

```
daily_intensity <- read_csv("C:\\Users\\laure\\OneDrive\\Documents\\Data Study\\Case Study\\Fitabase Data 4.12.16\\daily_intensity.csv")
```

```
## Rows: 934 Columns: 3
## -- Column specification -----
## Delimiter: ","
## dbl (2): Id, Total_intensity
## date (1): ActivityDate
##
## i Use 'spec()' to retrieve the full column specification for this data.
## i Specify the column types or set 'show_col_types = FALSE' to quiet this message.
```

2. We can see the error of date format of those datasets. Now, we proceed to change the format in “sleep” dataset.

```
sleep$SleepDay = as.POSIXct(sleep$SleepDay, format = "%m/%d/%Y %I:%M:%S %p", tz = Sys.timezone())
sleep$time <- format(sleep$SleepDay, format = "%H:%M:%S")
sleep$date <- format(sleep$SleepDay, format = "%Y-%m-%d")
head(sleep)
```

```
## # A tibble: 6 x 7
##       Id SleepDay      TotalSleepRecords TotalMi~1 Total~2 time  date
##       <dbl> <dtm>          <dbl>      <dbl>    <dbl> <chr> <chr>
## 1 1503960366 2016-04-12 00:00:00          1        327    346 00:0~ 2016~
## 2 1503960366 2016-04-13 00:00:00          2        384    407 00:0~ 2016~
## 3 1503960366 2016-04-15 00:00:00          1        412    442 00:0~ 2016~
## 4 1503960366 2016-04-16 00:00:00          2        340    367 00:0~ 2016~
## 5 1503960366 2016-04-17 00:00:00          1        700    712 00:0~ 2016~
## 6 1503960366 2016-04-19 00:00:00          1        304    320 00:0~ 2016~
## # ... with abbreviated variable names 1: TotalMinutesAsleep, 2: TotalTimeInBed
```

Then, we proceed to change the format in “daily\_intensity” and “hourly\_intensity” dataset.

```
daily_intensity$ActivityDate <-strptime(daily_intensity$ActivityDate, format = "%Y-%m-%d")
format(daily_intensity$ActivityDate, format = "%Y-%m-%d")
```

```
## [1] "2016-04-12" "2016-04-13" "2016-04-14" "2016-04-15" "2016-04-16"
## [6] "2016-04-17" "2016-04-18" "2016-04-19" "2016-04-20" "2016-04-21"
## [11] "2016-04-22" "2016-04-23" "2016-04-24" "2016-04-25" "2016-04-26"
## [16] "2016-04-27" "2016-04-28" "2016-04-29" "2016-04-30" "2016-05-01"
## [21] "2016-05-02" "2016-05-03" "2016-05-04" "2016-05-05" "2016-05-06"
## [26] "2016-05-07" "2016-05-08" "2016-05-09" "2016-05-10" "2016-05-11"
## [31] "2016-04-12" "2016-04-13" "2016-04-14" "2016-04-15" "2016-04-16"
## [36] "2016-04-17" "2016-04-18" "2016-04-19" "2016-04-20" "2016-04-21"
## [41] "2016-04-22" "2016-04-23" "2016-04-24" "2016-04-25" "2016-04-26"
## [46] "2016-04-27" "2016-04-28" "2016-04-29" "2016-04-30" "2016-05-01"
## [51] "2016-05-02" "2016-05-03" "2016-05-04" "2016-05-05" "2016-05-06"
## [56] "2016-05-07" "2016-05-08" "2016-05-09" "2016-05-10" "2016-05-11"
## [61] "2016-05-12" "2016-04-12" "2016-04-13" "2016-04-14" "2016-04-15"
## [66] "2016-04-16" "2016-04-17" "2016-04-18" "2016-04-19" "2016-04-20"
## [71] "2016-04-21" "2016-04-22" "2016-04-23" "2016-04-24" "2016-04-25"
## [76] "2016-04-26" "2016-04-27" "2016-04-28" "2016-04-29" "2016-04-30"
## [81] "2016-05-01" "2016-05-02" "2016-05-03" "2016-05-04" "2016-05-05"
## [86] "2016-05-06" "2016-05-07" "2016-05-08" "2016-05-09" "2016-05-10"
## [91] "2016-05-11" "2016-04-12" "2016-04-13" "2016-04-14" "2016-04-15"
## [96] "2016-04-16" "2016-04-17" "2016-04-18" "2016-04-19" "2016-04-20"
## [101] "2016-04-21" "2016-04-22" "2016-04-23" "2016-04-24" "2016-04-25"
## [106] "2016-04-26" "2016-04-27" "2016-04-28" "2016-04-29" "2016-04-30"
## [111] "2016-05-01" "2016-05-02" "2016-05-03" "2016-05-04" "2016-05-05"
## [116] "2016-05-06" "2016-05-07" "2016-05-08" "2016-05-09" "2016-05-10"
## [121] "2016-05-11" "2016-05-12" "2016-04-12" "2016-04-13" "2016-04-14"
## [126] "2016-04-15" "2016-04-16" "2016-04-17" "2016-04-18" "2016-04-19"
## [131] "2016-04-20" "2016-04-21" "2016-04-22" "2016-04-23" "2016-04-24"
## [136] "2016-04-25" "2016-04-26" "2016-04-27" "2016-04-28" "2016-04-29"
```

[illegible]

[illegible]

```

## [681] "2016-04-15" "2016-04-16" "2016-04-17" "2016-04-18" "2016-04-19"
## [686] "2016-04-20" "2016-04-21" "2016-04-22" "2016-04-23" "2016-04-24"
## [691] "2016-04-25" "2016-04-26" "2016-04-27" "2016-04-28" "2016-04-29"
## [696] "2016-04-30" "2016-05-01" "2016-05-02" "2016-05-03" "2016-05-04"
## [701] "2016-05-05" "2016-05-06" "2016-05-07" "2016-05-08" "2016-05-09"
## [706] "2016-05-10" "2016-05-11" "2016-05-12" "2016-04-12" "2016-04-13"
## [711] "2016-04-14" "2016-04-15" "2016-04-16" "2016-04-17" "2016-04-18"
## [716] "2016-04-19" "2016-04-20" "2016-04-21" "2016-04-22" "2016-04-23"
## [721] "2016-04-24" "2016-04-25" "2016-04-26" "2016-04-27" "2016-04-28"
## [726] "2016-04-29" "2016-04-30" "2016-05-01" "2016-05-02" "2016-05-03"
## [731] "2016-05-04" "2016-05-05" "2016-05-06" "2016-05-07" "2016-04-12"
## [736] "2016-04-13" "2016-04-14" "2016-04-15" "2016-04-16" "2016-04-17"
## [741] "2016-04-18" "2016-04-19" "2016-04-20" "2016-04-21" "2016-04-22"
## [746] "2016-04-23" "2016-04-24" "2016-04-25" "2016-04-26" "2016-04-27"
## [751] "2016-04-28" "2016-04-29" "2016-04-30" "2016-05-01" "2016-05-02"
## [756] "2016-05-03" "2016-05-04" "2016-05-05" "2016-05-06" "2016-05-07"
## [761] "2016-05-08" "2016-05-09" "2016-05-10" "2016-05-11" "2016-05-12"
## [766] "2016-04-12" "2016-04-13" "2016-04-14" "2016-04-15" "2016-04-16"
## [771] "2016-04-17" "2016-04-18" "2016-04-19" "2016-04-20" "2016-04-21"
## [776] "2016-04-22" "2016-04-23" "2016-04-24" "2016-04-25" "2016-04-26"
## [781] "2016-04-27" "2016-04-28" "2016-04-29" "2016-04-30" "2016-05-01"
## [786] "2016-05-02" "2016-05-03" "2016-05-04" "2016-05-05" "2016-05-06"
## [791] "2016-05-07" "2016-05-08" "2016-05-09" "2016-05-10" "2016-05-11"
## [796] "2016-05-12" "2016-04-12" "2016-04-13" "2016-04-14" "2016-04-15"
## [801] "2016-04-16" "2016-04-17" "2016-04-18" "2016-04-19" "2016-04-20"
## [806] "2016-04-21" "2016-04-22" "2016-04-23" "2016-04-24" "2016-04-25"
## [811] "2016-04-26" "2016-04-27" "2016-04-28" "2016-04-29" "2016-04-12"
## [816] "2016-04-13" "2016-04-14" "2016-04-15" "2016-04-16" "2016-04-17"
## [821] "2016-04-18" "2016-04-19" "2016-04-20" "2016-04-21" "2016-04-22"
## [826] "2016-04-23" "2016-04-24" "2016-04-25" "2016-04-26" "2016-04-27"
## [831] "2016-04-28" "2016-04-29" "2016-04-30" "2016-05-01" "2016-05-02"
## [836] "2016-05-03" "2016-05-04" "2016-05-05" "2016-05-06" "2016-05-07"
## [841] "2016-05-08" "2016-05-09" "2016-05-10" "2016-05-11" "2016-05-12"
## [846] "2016-04-12" "2016-04-13" "2016-04-14" "2016-04-15" "2016-04-16"
## [851] "2016-04-17" "2016-04-18" "2016-04-19" "2016-04-20" "2016-04-21"
## [856] "2016-04-22" "2016-04-23" "2016-04-24" "2016-04-25" "2016-04-26"
## [861] "2016-04-27" "2016-04-28" "2016-04-29" "2016-04-30" "2016-05-01"
## [866] "2016-05-02" "2016-05-03" "2016-05-04" "2016-05-05" "2016-05-06"
## [871] "2016-05-07" "2016-05-08" "2016-05-09" "2016-05-10" "2016-05-11"
## [876] "2016-04-12" "2016-04-13" "2016-04-14" "2016-04-15" "2016-04-16"
## [881] "2016-04-17" "2016-04-18" "2016-04-19" "2016-04-20" "2016-04-21"
## [886] "2016-04-22" "2016-04-23" "2016-04-24" "2016-04-25" "2016-04-26"
## [891] "2016-04-27" "2016-04-28" "2016-04-29" "2016-04-30" "2016-05-01"
## [896] "2016-05-02" "2016-05-03" "2016-05-04" "2016-05-05" "2016-05-06"
## [901] "2016-05-07" "2016-05-08" "2016-05-09" "2016-04-12" "2016-04-13"
## [906] "2016-04-14" "2016-04-15" "2016-04-16" "2016-04-17" "2016-04-18"
## [911] "2016-04-19" "2016-04-20" "2016-04-21" "2016-04-22" "2016-04-23"
## [916] "2016-04-24" "2016-04-25" "2016-04-26" "2016-04-27" "2016-04-28"
## [921] "2016-04-29" "2016-04-30" "2016-05-01" "2016-05-02" "2016-05-03"
## [926] "2016-05-04" "2016-05-05" "2016-05-06" "2016-05-07" "2016-05-08"
## [931] "2016-05-09" "2016-05-10" "2016-05-11" "2016-05-12"

```

```

hourly_intensity$ActivityDate <- strptime(hourly_intensity$ActivityDate, format = "%m/%d/%Y")
format(hourly_intensity$ActivityDate, format = "%Y-%m-%d")

```

[illegible]

[illegible]



[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]



[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]



[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]



[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]



[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]



[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]



[illegible]



































[illegible]

[illegible]

[illegible]



[illegible]









[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]





[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]



[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]



[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]



[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]



[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]



[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]



[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]



[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]



[illegible]

[illegible]

[illegible]

[illegible]

```
## [22063] "02:00:00" "03:00:00" "04:00:00" "05:00:00" "06:00:00" "07:00:00"
## [22069] "08:00:00" "09:00:00" "10:00:00" "11:00:00" "12:00:00" "13:00:00"
## [22075] "14:00:00" "15:00:00" "16:00:00" "17:00:00" "18:00:00" "19:00:00"
## [22081] "20:00:00" "21:00:00" "22:00:00" "23:00:00" "00:00:00" "01:00:00"
## [22087] "02:00:00" "03:00:00" "04:00:00" "05:00:00" "06:00:00" "07:00:00"
## [22093] "08:00:00" "09:00:00" "10:00:00" "11:00:00" "12:00:00" "13:00:00"
## [22099] "14:00:00"
```

```
head(hourly_intensity)
```

```
## # A tibble: 6 x 5
##       Id ActivityDate      ActivityHour TotalIntensity AverageIntensity
##       <dbl> <dtm>          <time>          <dbl>          <dbl>
## 1 1503960366 2016-04-12 00:00:00 00:00             20           0.333
## 2 1503960366 2016-04-12 00:00:00 01:00              8           0.133
## 3 1503960366 2016-04-12 00:00:00 02:00              7           0.117
## 4 1503960366 2016-04-12 00:00:00 03:00              0              0
## 5 1503960366 2016-04-12 00:00:00 04:00              0              0
## 6 1503960366 2016-04-12 00:00:00 05:00              0              0
```

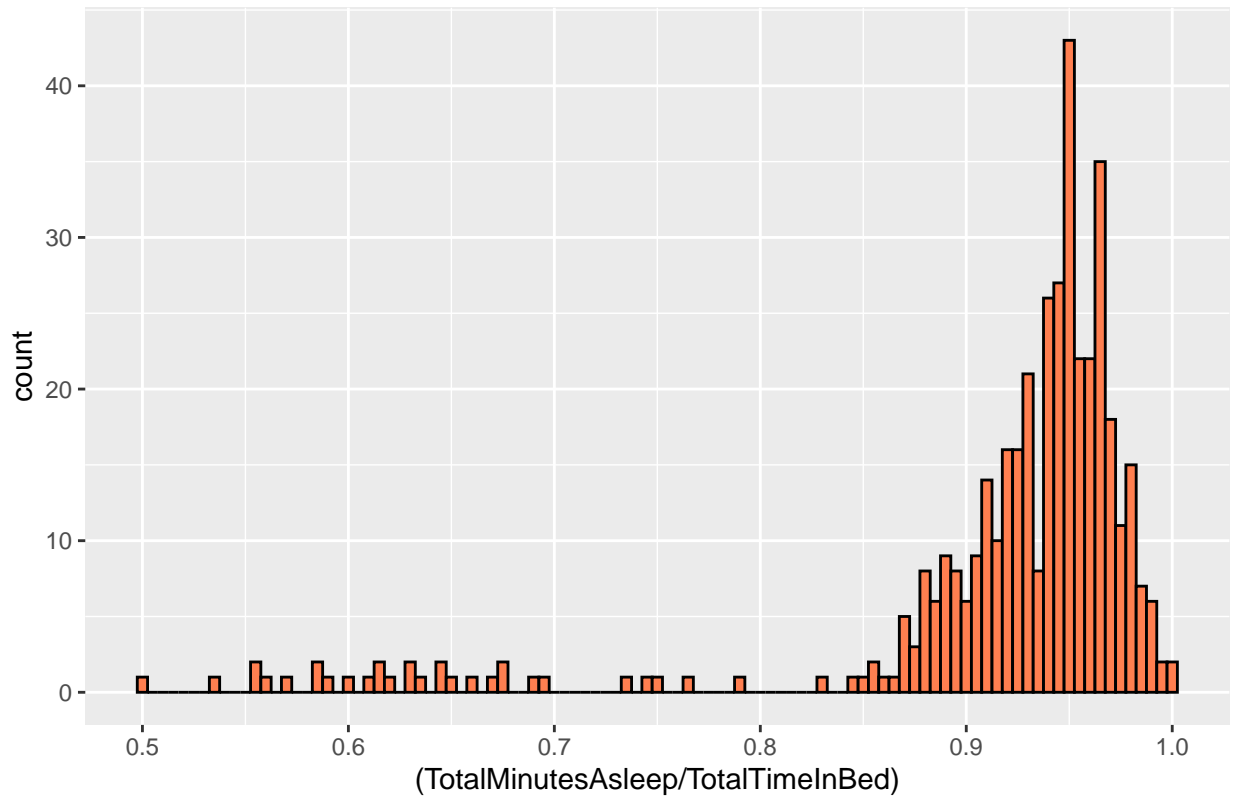
```
head(daily_intensity)
```

```
## # A tibble: 6 x 3
##       Id ActivityDate      Total_intensity
##       <dbl> <dtm>          <dbl>
## 1 1503960366 2016-04-12 00:00:00           429
## 2 1503960366 2016-04-13 00:00:00           318
## 3 1503960366 2016-04-14 00:00:00           293
## 4 1503960366 2016-04-15 00:00:00           364
## 5 1503960366 2016-04-16 00:00:00           349
## 6 1503960366 2016-04-17 00:00:00           318
```

3. We are checking how correlated between sleep time and total time in bed.

```
ggplot(data=sleep, mapping=aes(x=(TotalMinutesAsleep/TotalTimeInBed))) + geom_histogram(color='black',
```

Percentage of Sleep over Time In Bed



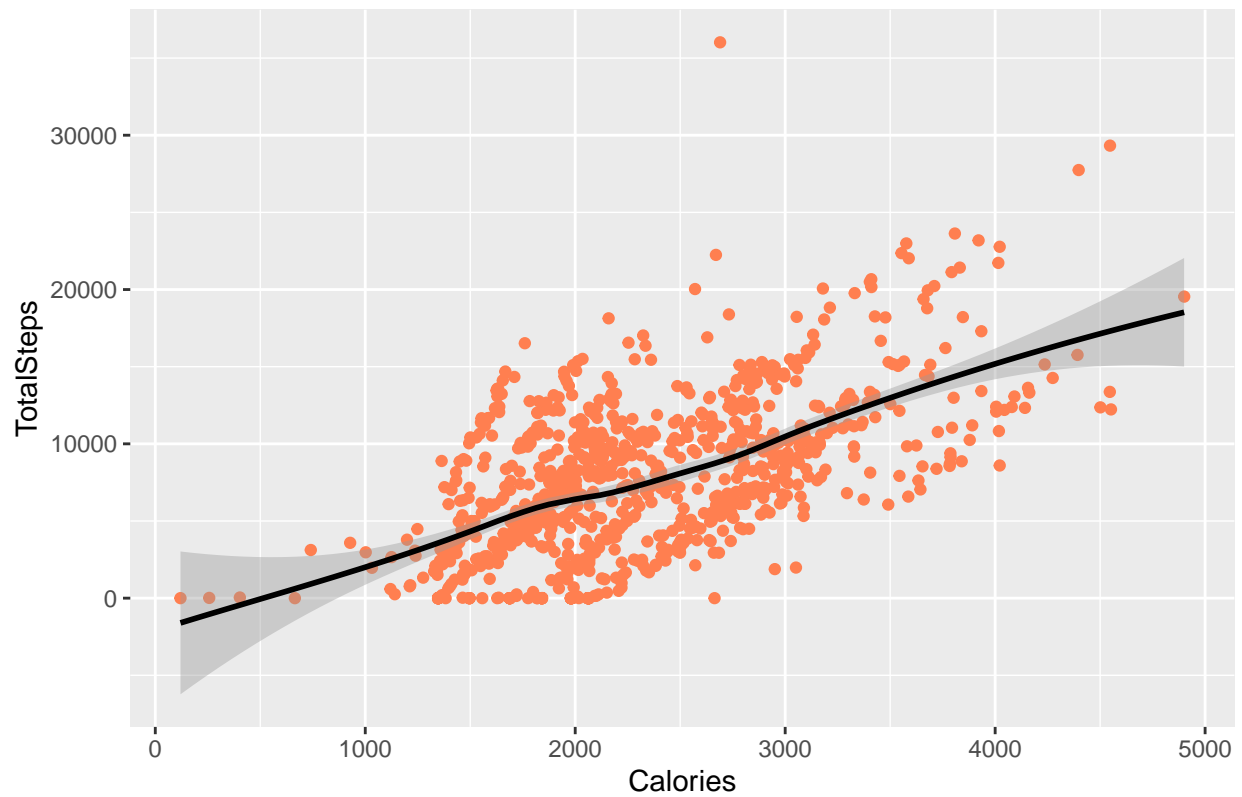
According to the chart, the most percentage people in sleep mode is 95% of time in bed. Almost people fall asleep fast when they are in bed, because the histogram is inclined to the range from 0.94 to 1.0.

4. We draw visualizations to compare effect of Distance, Steps, and Intensity on burning calories.

```
ggplot(data=activity, mapping=aes(x = Calories, y = TotalSteps)) + geom_jitter(color = 'coral') + geom_s
```

```
## 'geom_smooth()' using method = 'loess' and formula = 'y ~ x'
```

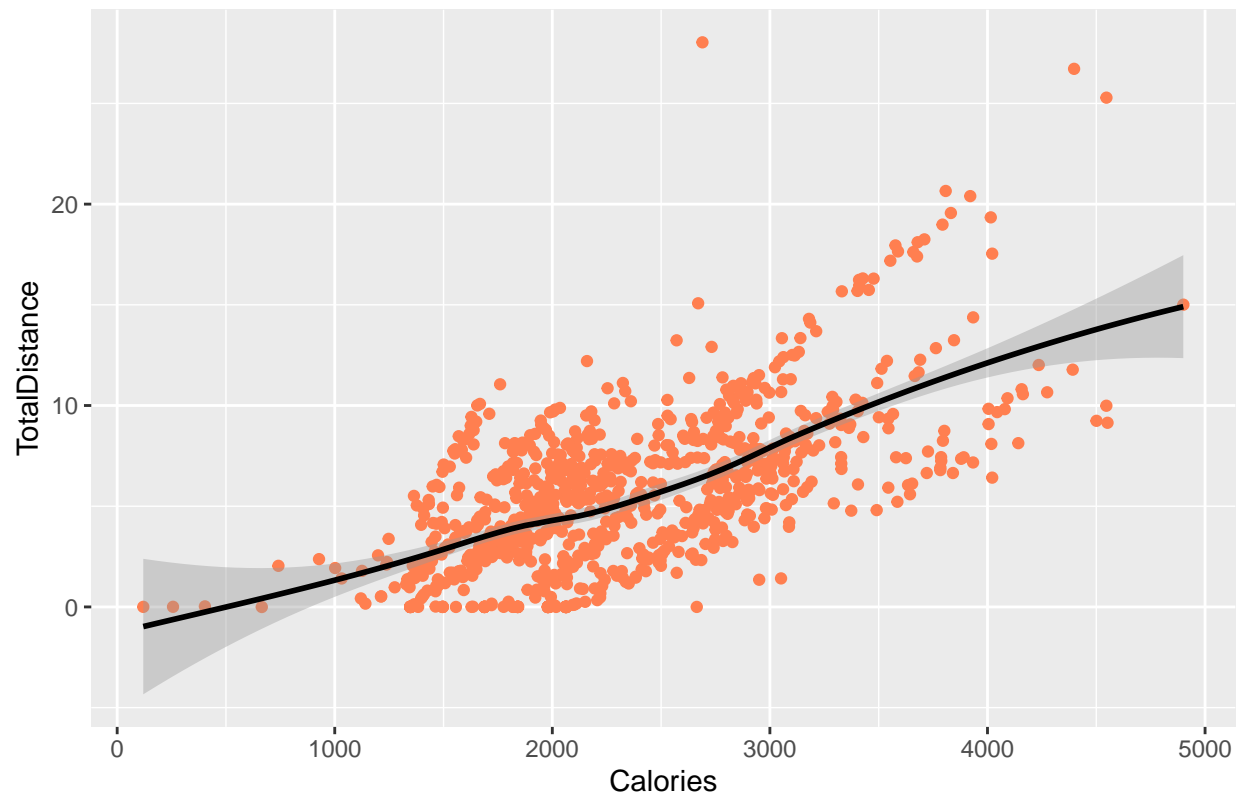
Total Steps vs. Calories Burnt



```
ggplot(data=activity, mapping=aes(x = Calories, y = TotalDistance)) + geom_jitter(color = 'coral') +geom
```

```
## 'geom_smooth()' using method = 'loess' and formula = 'y ~ x'
```

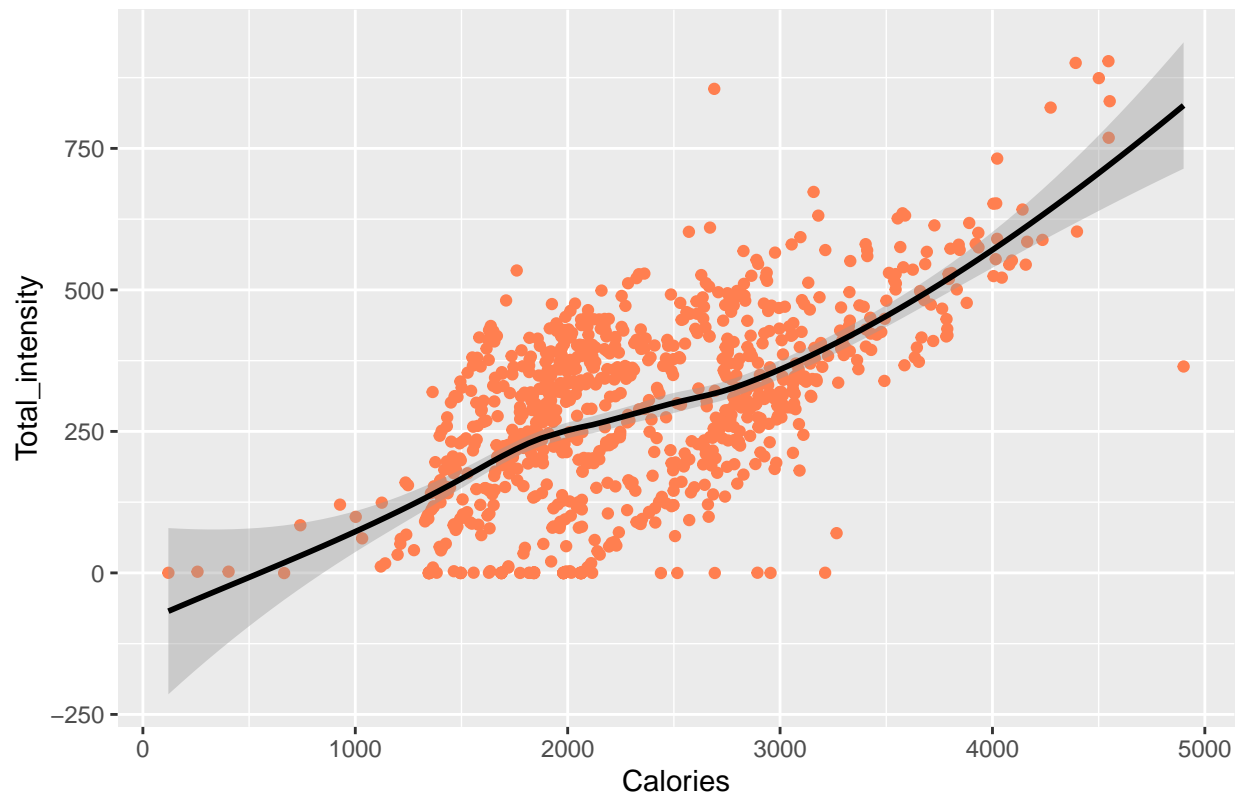
Total Distance vs. Calories Burnt



```
ggplot(data=activity, mapping=aes(x = Calories, y = Total_intensity)) + geom_jitter(color = 'coral') +  
  
## 'geom_smooth()' using method = 'loess' and formula = 'y ~ x'
```



## Total Intensity vs. Calories Burnt



As we can see, all of them have positive effect on calories. When step and distance are quite close to each other in effectiveness, the best way to burn out calories among those three is intensity. So, the rank from most to least effective element is intensity, distance and step.

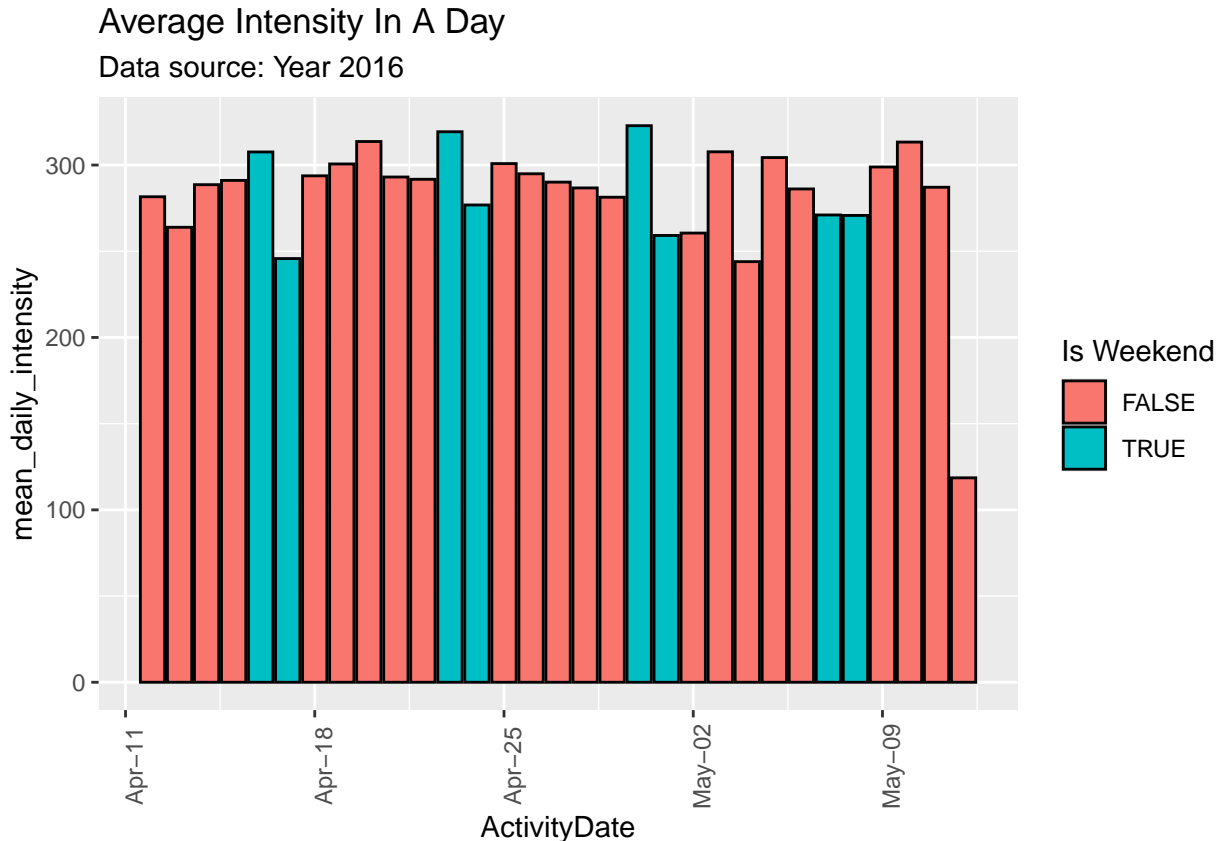
5. Analyze how people are active in days within a week. First, we create new data frame to customize the data we need.

```
sumdaily_intensity <- daily_intensity %>%
  group_by(ActivityDate) %>%
  drop_na() %>%
  summarise(mean_daily_intensity = mean(Total_intensity))
sumdaily_intensity$ActivityDate <- as.Date(sumdaily_intensity$ActivityDate)
```

Second, we create the visualization.

```
ggplot(sumdaily_intensity, mapping = aes(x = ActivityDate, y = mean_daily_intensity, fill=weekdays(ActivityDate)))
```

```
## Warning in geom_histogram(stat = "identity", color = "black"): Ignoring unknown
## parameters: 'binwidth', 'bins', and 'pad'
```



According to the visualization, we realize people maintain the activeness within a week gradually. They have the most intensive activities on Saturdays. And they usually take break on Sundays.

## Recommendations

Our target customer is young people (25-35 years old) who have awareness of body and health conditions. They tend to care of their diet, calories, fitness and sleep quality.

Based on the analysis above, there are some key points that we can adapt to company’s marketing strategy.

- \* We can recommend customers, who desire to increase calories burnt, to focus on vigorous exercise intensity, while still depends on their heart rate and health condition.
- + We need a detailed research on which exercise level of intensity is suitable for them.
- \* We can develop a function on the app and “Time” watch to remind customers to go to bed early, as the sleeping time and time in bed is highly correlated.
- + Meanwhile, we need more data about sleep quality, such as how deep their sleep is versus total asleep minutes.
- \* As customers intend to be more active on Saturday, and relax on Sunday, we can customize notifications reminding them to enjoy exercise activities on Saturday and rest on the next day.
- + For better conclusion, we should collect longer data with more days recorded.

Thanks for your reading.